Ref 1

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－285216

(43)公開日　平成10年(1998)10月23日

(51)Int.Cl.*	識別記号	ＦＩ	
Ｈ０４Ｌ　12/56		Ｈ０４Ｌ　11/20	１０２Ｚ
Ｇ０６Ｆ　13/00	３５７	Ｇ０６Ｆ　13/00	３５７Ｚ

審査請求　未請求　請求項の数6　ＯＬ　（全 19 頁）

(21)出願番号　特願平10－64914

(22)出願日　平成10年(1998)３月16日

(31)優先権主張番号　０８／８２８４４９

(32)優先日　1997年３月28日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

(72)発明者　ジョーゼフ・エム・クライトン
アメリカ合衆国12603　ニューヨーク州ポーキープシー　エドガー・ストリート　9

(74)代理人　弁理士　坂口　博　（外１名）

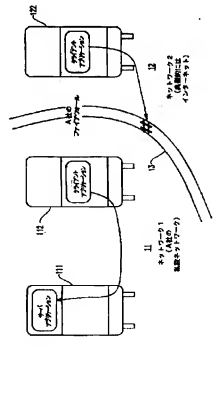
最終頁に続く

(54)【発明の名称】　セキュリティ保護通信トンネリングの方法及び装置

(57)【要約】

【課題】　軽セキュリティ保護トネリング・プロトコル（ＬＳＴＰ）によって、中間サーバまたはプロキシを使用して１つまたは複数のファイアウォールを通る通信を可能にする。

【解決手段】　ファイアウォールを通ってナビゲートする終端間接続を、３つのプロキシを使用して確立する。典型的な構成では、第１のファイアウォールの背後にあるサーバと第２のファイアウォールの背後にあるクライアントが、両ファイアウォール間の非信頼ネットワーク（たとえばインターネット）によって相互接続される。第１のファイアウォール内部ＳＯＣＫＳ認識サーバ側終端プロキシが、第１のファイアウォールの内部にあるサーバに接続する。第２のファイアウォール内部ＳＯＣＫＳ認識クライアント側終端プロキシが、第２のファイアウォールの内部にあるクライアントによって接続される。サーバ側とクライアント側の両方の終端プロキシが、２つのファイアウォールの外部にある第３のプロキシ（中間プロキシと呼ぶ）をアドレスすることができる。



1

【特許請求の範囲】

【請求項１】サーバ・アプリケーションを稼動させる少なくとも１つのサーバを含む第１のネットワークと、クライアント・アプリケーションを稼動させる少なくとも１つのクライアントを含む第２のネットワークと、第１のネットワークと第２のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第１のファイアウォールが第１のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む、第１のファイアウォールと、

相互にアドレス可能なサーバ終端プロキシ及びサーバ・アプリケーションと、

相互にアドレス可能なクライアント終端プロキシ及びクライアント・アプリケーションと、

第１のファイアウォールの外部にあり、第１のネットワークと第２のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含み、サーバ終端プロキシとクライアント終端プロキシがそれぞれ第１のファイアウォールを介して中間プロキシとの接続を行い、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバス・スルー通信トンネルを確立する、パケット交換ネットワーク通信システム。

【請求項２】第１のネットワークと第２のネットワークのうちの他方のネットワークのコンピュータ資源を保護し、第２のファイアウォールが第２のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む第２のファイアウォールをさらに含む、請求項１に記載のパケット交換ネットワーク通信システム。

【請求項３】サーバ終端プロキシと、クライアント終端プロキシと、中間プロキシがＳＯＣＫＳサーバ機能を有するトンネルを構成し、トンネル全体がＳＯＣＫＳサーバのジョブを実行する、請求項２に記載のパケット交換ネットワーク通信システム。

【請求項４】サーバ・アプリケーションを稼動させる少なくとも１つのサーバを含む第１のネットワークと、クライアント・アプリケーションを稼動させる少なくとも１つのクライアントを含む第２のネットワークと、第１と第２のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第１のファイアウォールが第１のファイアウォールの内部から外部に接続を行うことができるようにするソフトウェア・アプリケーションを含む第１のファイアウォールと、サーバ・アプリケーションによるアドレスが可能なサーバ終端プロキシと、クライアント・アプリケーションによるアドレスが可能なクライアント終端プロキシと、第１のファイアウォールの外部にあり、第１のネットワークと第２のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含むパケット交換ネットワーク通信システムにおいて、第１のファイアウォールを介して中間プロキシにサーバ終端プロキシとクライアント終端プロキシとを接続し、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバススルー通信トンネルを確立する方法であって、

中間プロキシを始動させ、終端プロキシからの第１の接続を待つステップと、

クライアント終端プロキシを始動させ、中間プロキシにクライアント・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、

中間プロキシによって、終端プロキシ・セットアップ情報を記憶し、その後で第２の接続を待つステップと、

サーバ終端プロキシを始動させ、中間プロキシに終端プロキシ・セットアップ情報を送信することによって中間プロキシとの接続を開くステップと、

中間プロキシによって、クライアント終端プロキシの接続とサーバ終端プロキシの接続とを対にし、サーバ及び中間プロキシ・セットアップ情報をクライアント終端プロキシに送信し、クライアント及び中間プロキシ・セットアップ情報をサーバ終端プロキシに送信するステップと、

その後で中間プロキシがクライアント終端プロキシとサーバ終端プロキシとの間のバス・スルーとして機能するステップとを含む方法。

【請求項５】中間プロキシによってクライアント終端プロキシとサーバ終端プロキシの間の接続を対にした後で、クライアント終端プロキシとサーバ終端プロキシによってセキュリティ・ハンドシェークを交換するステップと、

セキュリティ保護された回線で中間プロキシを介してクライアント終端プロキシとサーバ終端プロキシの間でセットアップ情報を再び交換するステップとをさらに含む、請求項４に記載の方法。

【請求項６】トンネルを介したクライアントとサーバの間のデータ交換が完了したときにクライアント終端プロキシとサーバ終端プロキシとの間のトンネル資源を解放するステップをさらに含む、請求項５に記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、一般にはパケット交換ネットワーク通信に関し、詳細には、外部クライアントが内部サーバにアドレスすることができないように組織の「ファイアウォール」が構成されている場合であっても、組織の「ファイアウォール」の外部にあるＴＣＰ／ＩＰクライアントがその同じファイアウォールの内部にあるサーバをアドレスすることができるようにする方法及び装置に関する。

【０００２】

【従来の技術】インターネットは、政府機関、教育機関、及び民間企業を含む参加組織の間での資源の共有を容易にする世界中のネットワークの集まりである。これらのネットワークは、転送制御プロトコル／インターネット・プロトコル（ＴＣＰ／ＩＰ）プロトコル群を使用し、共通アドレス空間を共用する。したがって、インターネット上のコンピュータは互換性のある通信標準を使用し、互いに連絡してデータを交換する能力を共用する。インターネットのユーザは主に、電子メール（ｅメール）や、ユーザがリモート・ホストにログ・インすることができるようにするプロセスであるＴｅｌｎｅｔや、リモート・ホスト上の情報をローカル・サイトに転送することができるようにするプロトコルであるファイル転送プロトコル（ＦＴＰ）の実現を介して通信する。

【０００３】ローカル・エリア・ネットワーク（ＬＡＮ）などのネットワークをインターネットに接続する場合、セキュリティが大きな問題である。重要な問題の１つはローカル・ホストへのアクセスを獲得しようと試みる侵入者である。この種の侵入を防止する一般的な方法は、インターネットへのセキュリティ保護された単一の接続点である、いわゆる「ファイアウォール」を設置することである。この単一接続点は、ファイアウォール管理者によって指定された特定の通信のみを通過させるようファイアウォール・ホストの形態をとる。典型的なファイアウォール・ホスト実施態様では、ＬＡＮ内のホスト上のファイルをインターネットを介して外部ホストに転送しようとするユーザは、まずそのファイルをファイアウォール・ホストに転送し、次にファイアウォールにログ・インタフェースしてファイルを外部ホストに転送する。この手続きは単一ユーザに対しては高度なセキュリティを与えるが、そのホストへのアクセスを必要とするユーザの数が増えるにつれてセキュリティを維持するのが難しくなる。ファイアウォールの概要については、ウィリアムＲ．チェスウィック（Ｗｉｌｌｉａｍ Ｒ．Ｃｈｅｓｗｉｃｋ）及びスティーブン・Ｍ．ベロヴィン（Ｓｔｅｖｅｎ Ｍ．Ｂｅｌｌｏｖｉｎ）著の「Ｆｉｒｅｗａｌｌｓ ａｎｄ Ｉｎｔｅｒｎｅｔ Ｓｅｃｕｒｉｔｙ」（Ａｄｄｉｓｏｎ－Ｗｅｓｌｅｙ，１９９４）等に記載されている。

【０００４】セキュリティの問題を最小限にすると同時に多数のユーザによるアクセスを可能にすることを目指して、ＳＯＣＫＳと呼ばれるトランスポート層プロキシ・アーキテクチャが作成された。これについては、たとえばデイヴィッド・コブラス（Ｄａｖｉｄ Ｃｏｂｌａｓ）及びミシェルＲ．コブラス（ＭｉｃｈｅｌｌｅＲ．Ｋｏｂｌａｓ）の「ＳＯＣＫＳ」（ＵＮＩＸ Ｓｅｃｕｒｉｔｙ Ｓｙｍｐｏｓｉｕｍ，ＵＳＥＮＩＸ Ａｓｓｏｃｉａｔｉｏｎ，１９９２），７７～８３ページ、ならびにインダ・リー（Ｙｉｎｇ－Ｄａ Ｌｅｅ）の「ＳＯＣＫＳ：Ａｐｒｏｔｏｃｏｌ ｆｏｒ ＴＣＰ ｐｒｏｘｙ ａｃｒｏｓｓ ｆｉｒｅｗａｌｌｓ」（ｈｔｔｐ：／／ｗｗｗ．ｓｏｃｋｓ．ｎｅｃ．ｃｏｍ／ｓｏｃｋｓ４．ｐｒｏｔｏｃｏｌ），ならびにＭ．リーチ（Ｌｅｅｃｈ），Ｍ．ガンス（Ｇａｎｉｓ），Ｙ．リー（Ｌｅｅ），Ｒ．クリス（Ｋｕｒｉｓ），Ｄ．コブラス（Ｋｏｂｌａｓ），及びＬ．ジョーンズ（Ｊｏｎｅｓ）の「ＳＯＣＫＳ Ｐｒｏｔｏｃｏｌ Ｖｅｒｓｉｏｎ５」（ｆｔｐ：／／ｄｓ．ｉｎｔｅｒｎｉｃ．ｎｅｔ／ｒｆｃ／ｒｆｃ１９２８．ｔｘｔ）等に記載されている。

トランスポート層プロキシ・アーキテクチャでは、ファイアウォールの背後にあるクライアントとよばれる１つの終端システムが、ファイアウォール上にあるとみなせるプロキシへの接続を行うことによってセッションを開始する。クライアントとプロキシはこの接続を使用してメッセージを交換し、認証やプロキシ要求などのセッション・セットアップ情報（たとえばファイアウォール・プロキシの場合には接続する外部ホスト、ＨＴＴＰ（ハイパーテキスト転送プロトコル）プロキシの場合にはフェッチするＵＲＬ（ユニフォーム・リソース・ロケータ）など）について折衝する。次に、プロキシは、通例は、クライアントの指示に従って、サーバと呼ばれる一般にはファイアウォールの外部にある別の終端システムへの接続を開く要求を行う。プロキシは、この接続を介してサーバとセッション・セットアップ情報を交換することができる。両方の接続でセッション・セットアップが完了した後、プロキシはこの２つの接続間で相互にデータのコピーを開始し、ホスト間を流れる情報の削除、追加、または変更は行わない（ただし、ＨＴＴＰキャッシュ・プロキシの場合のように情報のコピーを密かに保持することはできる）。

【０００５】組織内の従業員は、「外部」クライアントが自分の「内部」サーバをアドレスすることができるようにしたいと考えることが多い。この場合、従業員は外部クライアントを信頼しているため、ファイアウォール上に設置された制御を迂回して、信頼された外部クライアントが信頼された内部サーバをアドレスすることができるようにしたいと考えることがある。

【０００６】

【発明が解決しようとする課題】したがって、本発明の目的は、既存のファイアウォール・ソフトウェアやファイアウォール構成に変更を加えることなく、ファイアウォールを通過するセキュリティ保護されたトンネルを確立する方法を提供することである。

【０００７】本発明の他の目的は、チャネルを介した通信の終端間プライバシー及び保全性のための備えを提供することである。

【０００８】本発明の他の目的は、セキュリティ保護されたチャネルを確立するユーザの相互認証のための備えを提供することである。

【０００９】本発明の他の目的は、ユーザＩＤとパスワ

ードをリモート・サーバに私的送達することができるようにするログイン／パスワードに基づくプロトコルを提供することである。

【００１０】

【課題を解決するための手段】本発明によると、中間サーバまたはプロキシを使用して１つまたは複数のファイアウォールを通る通信を可能にする、「軽セキュリティ保護トンネリング・プロトコル」ＬＳＴＰ（Ｌｉｇｈｔ ｗｅｉｇｈｔ Ｓｅｃｕｒｅ Ｔｕｎｎｅｌｉｎｇ Ｐｒｏｔｏｃｏｌ）が提供される。具体的には、この基本システムは１つの中間プロキシ（ｍｉｄｄｌｅ ｐｒｏｘｙ）と２つの終端プロキシ（ｅｎｄｐｒｏｘｙ）から成る３つのプロキシを使用して、２つのファイアウォールを通過してナビゲートする終端間接続を確立する。この構成では、第１のファイアウォールの背後にあるサーバと第２のファイアウォールの背後にあるクライアントが２つのファイアウォール間にある信頼性のないネットワーク（たとえばインターネット）によって相互接続される。第１のファイアウォール内部ＳＯＣＫＳ認識終端サーバ側プロキシが、第１のファイアウォール内部サーバに接続する。第２のファイアウォール内部のクライアントは、第２のファイアウォール内部ＳＯＣＫＳ認識クライアント側終端プロキシに接続する。サーバ側とクライアント側の両方の終端プロキシは、この２つのファイアウォールの外部にある第３のプロキシ（中間プロキシと呼ぶ）をアドレスすることができる。他の２つのプロキシ（サーバとクライアントの終端プロキシ）が始動後しばらくしてから中間プロキシへの接続を開始するため、通常は中間プロキシが先に始動される。中間プロキシは、両方の内部終端プロキシによって相互にアドレス可能であるため、サーバとクライアントの間に完全な終端間接続が確立される。１つまたは複数の中間プロキシをＬＳＴＰのような適切なプロトコルと共に使用することで、複数のファイアウォールを通るセキュリティ保護された通信リンクまたはトンネルが確立される。

【００１１】

【発明の実施の形態】図面、特に図１を参照すると、ファイアウォール１３によって分離された２つのネットワーク１１及び１２が図示されている。ファイアウォール１３は、第１のネットワーク１１をファイアウォールの外部から起こされる不正活動から保護するために使用される。この図では、ネットワーク１１はＡ社の私設ネットワークであり、ネットワーク１２は典型的にはインターネットである。この例では、ネットワーク１１はサーバ・アプリケーションを稼働させているサーバ１１１と、クライアント・アプリケーションを稼働させているクライアント１１２によって表されている。ファイアウォール１３の背後にあるクライアント１１２上で稼働しているクライアント・アプリケーションは、サーバ１１１上で稼働しているサーバ・アプリケーションをアドレスすることができるが、ファイアウォール１３の外部にあるクライアント１１２上で稼働しているクライアント・アプリケーションはサーバ１１１上で稼働しているサーバ・アプリケーションをアドレスすることができない。これは、この通信がファイアウォールによって遮断されているためである。すなわち、ファイアウォール１３の目的は、この場合はローカル・エリア・ネットワーク（ＬＡＮ）上で互いに接続されている多くのサーバとクライアントを含むことができるネットワーク１１であるＡ社のコンピュータ資源を保護することである。

【００１２】図２に、ネットワーク２内でサーバ・アプリケーションを稼働させているサーバ１２１が存在する点以外は同様の配置に図示されている。図１と同様に、クライアント１１２上で稼働しているクライアント・アプリケーションはこの場合も、サーバ１１１上で稼働しているサーバ・アプリケーションをアドレスすることができる。内部から外部への通信ができるように使用可能になっていれば、ファイアウォール１３はクライアント１１２上で稼働しているクライアント・アプリケーションがファイアウォール１３の外部にあるサーバ１２１上で稼働しているサーバ・アプリケーションに接続することができるようにもする。前述のＳＯＣＫＳと呼ばれる共通ソフトウェア・パッケージによって、ファイアウォールは図２に示すように内部から外部への接続を行うことができる。

【００１３】図３にも２つのネットワークが示されているが、第２のネットワーク１４は今度は第２のファイアウォール１５の背後にあるＢ社の私設ネットワークである。したがって、ファイアウォール１５はＢ社のコンピュータ資源をファイアウォール１５の外部から起こされる不正な活動から保護するように設計されている。この図では、ファイアウォール１５の背後にあるクライアント１４２上で稼働しているクライアント・アプリケーションは、インターネット１２を介してファイアウォール１３の背後にあるサーバ１１１上で稼働しているサーバ・アプリケーションへのアドレスを試みることができる。しかし、ファイアウォール１５はクライアント１４２上で稼働しているクライアント・アプリケーションがファイアウォール１５の外部に接続することができるようにするＳＯＣＫＳ機能を備えることができるのに対し、ファイアウォール１３はサーバ１１１上で稼働しているサーバ・アプリケーションへの接続を禁止する。

【００１４】組織内部の従業員は「外部」クライアント・アプリケーションが「内部」サーバをアドレスすることができるようにしたいと考えることが多い。たとえば、図３のクライアント１４２上で稼働しているクライアント・アプリケーションがサーバ１１１上で稼働しているサーバ・アプリケーションをアドレスすることができるようにしたい場合がある。この場合、従業員は外部クライアント・アプリケーションを信頼し、信頼されて

いる外部クライアントが内部サーバをアドレスするのを禁止する自社のファイアウォールに設置された制御を迂回したいと考える。本発明は、このような状況の解決策を提供する。

【００１５】以下に、本発明について一般性を失うことなく、Ｘ Ｗｉｎｄｏｗｓ Ｓｙｓｔｅｍにおける実施態様に関して説明する。Ｘ Ｗｉｎｄｏｗｓ Ｓｙｓｔｅｍは、ＵＮＩＸワークステーション用にＭＩＴで開発され、ハードウェア独立グラフィカル・ユーザ・インタフェース（ＧＵＩ）の作成を可能にする表示処理ルーチンの標準セットである。最初に説明する例では、２つのファイアウォールを扱う。この最初の実施態様は、ＳＯＣＫＳゲートウェイ内部のＳＯＣＫＳ認識プログラムがＳＯＣＫＳゲートウェイ外部のサーバに接続することができるようにするＳＯＣＫＳパッケージを基に構築されている。

【００１６】図４に、本発明による、セキュリティ保護されたトンネルを構築するために使用する３つのタイプのプロキシとネットワーク構成におけるその配置とを示す。Ａ社の私設ネットワークまたはイントラネットであるネットワーク２１はファイアウォール２３によって保護され、Ｂ社の私設ネットワークまたはイントラネットであるネットワーク２２はファイアウォール２５によって保護されている。ファイアウォール２３の背後にはＡ社の私設ネットワークの一部であるＸサーバ２１１があり、ファイアウォール２５の背後にはＢ社の私設ネットワークの一部であるＸクライアント２２２がある。ファイアウォール２３と２５は両方とも、「内部」クライアントが「外部」サーバに接続することができるようにするＳＯＣＫＳ機能を備えている。Ｘクライアント２２２はＢ社のファイアウォール２５の背後にあるそのアドレス可能ドメイン内に、Ｘプロトコルの着信を監視することができるクライアント終端プロキシ２２３を持っている。クライアント終端プロキシ２２３は、Ｘクライアント２２２にとってローカルＸサーバであるように見えるため、Ｘクライアント２２２には修正を加える必要がない。

【００１７】Ｘサーバ２１１のアドレス可能ドメイン内にサーバ終端プロキシ２１３があるＡ社のファイアウォール２３の背後にも同様の状況が存在する。サーバ終端プロキシ２１３は、実Ｘクライアントとまったく同様にＸサーバ２１１に接続することができる。サーバ終端プロキシ２１３はＸサーバ２１１にとってローカルＸクライアントのように見えるため、この場合もＸサーバ２１１には修正を加える必要がない。

【００１８】終端プロキシが中間プロキシへの接続を開始することになるため、中間プロキシ２６が先に始動する。クライアント終端プロキシ２２３とサーバ終端プロキシ２１３は既存の機能（たとえばＳＯＣＫＳ）を使用して内部から外部にファイアウォールを通って要求を行う。中間プロキシ２６は（各ファイアウォール上のＳＯＣＫＳを使用して）両方の終端プロキシによって相互にアドレス可能であるため、中間プロキシ２６を介してＸクライアント２２２とＸサーバ２１１の間に完全な終端間接続を確立することができる。

【００１９】中間プロキシ２６はクライアント終端プロキシ２２３とサーバ終端プロキシ２１３の両方にとってサーバのように見え、本発明の重要な特徴である。したがって、クライアント終端プロキシ２２３とサーバ終端プロキシ２１３は両方ともそれぞれのファイアウォール２５及び２３を通過して中間プロキシ２６をアドレスすることができる。カスケード中間プロキシまたは複数の中間プロキシの場合、中間プロキシは終端プロキシによってアドレスされるのではなく、実際に他の中間プロキシをアドレスする。標準ＴＣＰ／ＩＰ接続機構を使用して初期接続が行われる。確立された各接続は、どのプログラムによって開始されたかを問わず、ＴＣＰ／ＩＰ接続であり、したがって全二重である。本発明は、ＴＣＰ／ＩＰ上で使用してトンネル管理事象の適切なセキュリティ保護を行う「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）を提供する。ＬＳＴＰは、トンネル構築中だけでなくトンネル存続期間全体を通じて、クライアント終端プロキシと、サーバ終端プロキシと、中間プロキシの間で「使われる」。

【００２０】終端プロキシ２１３と２２３が中間プロキシ２６への接続を開始するトリガは、終端プロキシ２１３及び２２３が稼動しているコンピュータへのアクセス権を持っている人によって主導制御される。終端プロキシ２１３及び２２３は、中間プロキシが始動された後いつでも中間プロキシ２６との接続を確立することができる。中間プロキシ２６は最初に接続する終端プロキシから送られたセットアップ情報を受信し、記憶する。

【００２１】中間プロキシが、クライアント終端プロキシ２２３からの接続とサーバ終端プロキシ２１３からの接続の２つの一致する接続を持っている場合、中間プロキシ２６はその２つの接続に加わり、透明なパイプのような役割を果たして、実質的に２つの終端プロキシの間に接続を確立する。この時点以降、中間プロキシ２６はパス・スルー・モードになり、一方の終端プロキシが他方の終端プロキシとのセキュリティ・ハンドシェークを開始してこのトンネルをセキュリティ保護する。

【００２２】これでＸクライアント２２２はクライアント終端プロキシ２２３と接続することができ、Ｘサーバ２１１に直接接続されているかのように終端プロキシ２２３にデータを渡す。Ｘクライアント２２２からの初期データによって、２つの終端プロキシ２２３及び２１３と中間プロキシ２６によって確立されたトンネルを通ってＬＳＴＰメッセージが流れ、次にそれによってサーバ終端プロキシ２１３がＸサーバ２１１との接続を開始する。その後、Ｘクライアント２２２からのデータがトン

ネルを通過し、Xサーバ２１１に渡される。互いに直接通信していないことを示すものは、Xクライアント２２２にもXサーバ２１１にもまったくない。データは、Xサーバ２１１からXクライアント２２２へと、Xクライアント２２２からXサーバ２１１への両方向に流れる。この時点で、追加のクライアントがクライアント・プロキシに接続して同じトンネル接続を使用したり、新しいトンネル接続を要求したりすることができる。

【００２３】要約すると、サーバ側終端プロキシは内部X Windows Systemサーバ及び外部中間プロキシと接続することができ、X Windows Systemクライアントはクライアント側終端プロキシと接続することができ、さらにクライアント側終端プロキシがXクライアントのために外部中間プロキシと接続することができる。確立される接続が本質的に全二重であるためと、過度的な閉鎖状態のため、X Windows Systemクライアントはファイアウォールがないかのように X Windows Systemサーバをアドレスすることができる（すなわち、同じアドレス可能ネットワーク上にあるかのようである）。当業者なら、他のプロトコルやサービスに対応できるように終端プロキシの機能を増強することができることがわかるであろう。たとえば、１つの終端プロキシがクライアントとサーバの両方の終端プロキシ機能を備えることもできる。

【００２４】図５は、クライアント・プロキシと中間プロキシとサーバ・プロキシとの間の対話を示すデータ流れ図である。このプロセスは、中間プロキシが始動されており、最初の接続を待っている状態を仮定している。クライアント終端プロキシが始動され、中間プロキシとの接続を開き、中間プロキシにクライアント・セットアップ情報を送る。「セットアップ情報」とは、２つのLSTPプロトコルを説明する一般用語である。中間プロキシは終端プロキシ・セットアップ情報を記憶した後、２番目の接続を待つ。サーバ終端プロキシが始動され、中間プロキシとの接続を開き、サーバ終端プロキシが中間プロキシに終端プロキシ・セットアップ情報を送る。中間プロキシはこの２つの接続を対応し、クライアント終端プロキシにサーバ・プロキシと中間プロキシのセットアップ情報を送信し、サーバ終端プロキシにクライアント・プロキシと中間プロキシのセットアップ情報を送信する。

【００２５】プロセスのこの時点で、中間プロキシは接続における積極的な役割を終え、クライアント終端プロキシとサーバ終端プロキシとの間のパス・スルーの役割を果たす。接続が確立された後は、いずれか一方の終端プロキシからセキュリティ・ハンドシェークを開始することができる。各終端プロキシ内のアルゴリズムがセットアップ情報を使用してどちらの終端プロキシがセキュリティ・ハンドシェークを開始するかを決定する。図５には、クライアント終端プロキシがセキュリティ・ハンドシェークを送信し、それがサーバ終端プロキシに渡されるように図示されている。サーバ終端プロキシは、セキュリティ・ハンドシェークで応答し、それがクライアント終端プロキシに渡される。このセキュリティ・プロトコルが遂行されると、セキュリティ保護された回線を介して追加のセットアップ情報が再送信され、トンネル構築が完了する。当業者なら、トンネルを介した終端間セキュリティを確立するために他の事象シーケンスも使用可能であることがわかるであろう。クライアント・アプリケーションとサーバ・アプリケーションとの接続が完了すると、後述の「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）などのプロトコルを使用して２つのアプリケーションの間で双方向にデータを安全に受け渡しすることができる。その結果、クライアント・アプリケーションとサーバ・アプリケーションはそれぞれのアドレス可能性を実質的に拡張させたことになる。

【００２６】サーバ・プロキシ２１３とクライアント・プロキシ２２３は、（１）認証、暗号化、及び保全と、（２）ファイアウォールの通過と、（３）データ圧縮とを扱う。中間プロキシ２６は双方向パイプとして機能する。サーバ終端プロキシ２１３はXサーバ２１１にとってXクライアントのように見え、クライアント終端プロキシ２２３はXクライアント２２２にとってXサーバのように見える。サーバ終端プロキシ及びクライアント終端プロキシは通常、それぞれXサーバ及びXクライアントと同じマシン上にある。

【００２７】この概説では、２つのネットワーク２１と２２の間にセキュリティ保護トンネルを実現するためにいくつかの実施態様上の決定を行った。第１に、この例では、トンネル両端でX Windows System クライアントとサーバを使用した。したがって、終端プロキシはXプロトコルの着信を監視し、それに応答するようにカスタマイズされている。第２に、トンネルをセキュリティ保護するセキュリティ・プロトコルとしてセキュア・ソケット層（ＳＳＬ）を選定した。ＳＳＬは、発信側のデータ保全、データ・プライバシー、及び認証を行う。第３に、プロキシがファイアウォールの内部からファイアウォールの外部への接続を確立することができるようにする機構としてＳＯＣＫＳを選定した。第４に、トンネル構築、管理、データ・フロー制御、及びトンネル破棄を定式化する手段を提供するために、トンネルと共に「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）を開発した。

【００２８】本発明による「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）は、図４に示すように、クライアント・プロキシ２２３と中間プロキシ２６との間と、サーバ・プロキシ２１３と中間プロキシ２６との間で使用するプロトコルである。好ましい実施例で

は、ＬＳＴＰはプロキシ間のデータ転送とプロキシの状態の同期化のために、要求と応答について以下のような意味と順序づけ規則を含む。

ＤＯＭＭＯＮＵＮＩＱＵＥ＿ＩＮＦＯ － この情報によって２つの終端プロキシが同じ中間プロキシに異なる時点で接続することができる。両方のプロキシが同じ共通情報を提供したため、中間プロキシはこの２つの終端プロキシを互いに対にすべきであることを認識する。固有情報を使用して各終端プロキシ・ユーザを識別することもできる。

ＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥ － これはトンネルのトポロジを記述する情報である。中間プロキシは受け取ったどのＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥにもそのトポロジ（たとえば名前，アドレスなど）情報を付加し、そのＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥを終端プロキシまで転送する。これによって、各終端プロキシに、どのプロキシがトンネルに関与しているかを示すマップが提供される。

ＰＲＯＰＥＲＴＹ＿ＥＸＣＨＡＮＧＥ － これは，終端プロキシがそれ自体に関する情報を交換することができるようにする機構である。

ＣＯＮＮＥＣＴＩＯＮ＿ＲＥＱＵＥＳＴ － この要求によって一方の終端プロキシが他方の終端プロキシに対して，クライアント・アプリケーションがクライアント・アプリケーションによって使用されるトンネル資源の割り振りを要求していることを通知することができる。要求される資源には，既存のトンネル接続上の「多重化」チャネルまたは確立済みトンネル接続に追加される新規トンネル接続が含まれる。

ＣＯＮＮＥＣＴＩＯＮ＿ＡＣＫ及びＣＯＮＮＥＣＴＩＯＮ＿ＮＡＣＫ － この２つの応答によって，ＣＯＮＮＥＣＴＩＯＮ＿ＲＥＱＵＥＳＴを受け取るプロキシはトンネル資源を求める要求を受け入れまたは拒否することができる。

ＳＥＲＶＩＣＥ＿ＢＥＧＩＮ＿ＲＥＱＵＥＳＴ － この要求によって，終端プロキシは，アプリケーションがデータの送信を開始し，したがって要求または割振りされたトンネル資源の使用を開始することを他方の終端プロキシに通知することができる。

ＳＥＲＶＩＣＥ＿ＢＥＧＩＮ＿ＡＣＫ及びＳＥＲＶＩＣＥ＿ＢＥＧＩＮ＿ＮＡＣＫ － この２つの応答によって，終端プロキシは割り振られている資源の使用を開始するアプリケーションの要求を受け入れまたは拒否することができる。

ＳＥＲＶＩＣＥ＿ＤＡＴＡ － このメッセージは，２つの終端プロキシ間でアプリケーション・データを送信するために使用される。各クライアント／サーバ・アプリケーションの対がその対のＳＥＲＶＩＣＥ＿ＤＡＴＡメッセージ内に固有の識別子を含んでおり、それによって複数のアプリケーションが１つのＴＣＰ／ＩＰ接続を介してそれらのデータを多重化することができる。

ＳＥＲＶＩＣＥ＿ＤＡＴＡ＿ＰＡＵＳＥ － このメッセージによって，終端プロキシは他方の終端プロキシにアプリケーション・データの送信を停止するように指示することができる。

ＳＥＲＶＩＣＥ＿ＤＡＴＡ＿ＲＥＳＵＭＥ － このメッセージによって終端プロキシは他方の終端プロキシにアプリケーション・データの送信を再開するように指示することができる。

ＳＥＲＶＩＣＥ＿ＦＲＥＥ＿ＲＥＱＵＥＳＴ － この要求によって，終端プロキシは他方の終端プロキシに、アプリケーションが完了し、トンネル資源を解放することができることを通知することができる。

ＳＨＵＴＤＯＷＮ － これによって，終端プロキシは他方の終端プロキシに通知することによってトンネルを正常に遮断することができる。

ＥＲＲＯＲ － このメッセージによって，終端プロキシはエラー情報を交換することができる。

【００２９】ＬＳＴＰを図６にまとめる。最初の３つのメッセージはトンネルのセットアップと管理を扱うために使用されることに留意されたい。具体的には，セットアップ情報は，ＣＯＭＭＯＮＵＮＩＱＵＥ＿ＩＮＦＯとＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥ ＬＳＴＰメッセージの様々な組み合わせからなり，トンネル接続を扱うために使用される。次の６つのメッセージは，トンネルを使用するアプリケーションの必要に応じて既存のトンネルの資源を扱うために使用される。その次のメッセージＳＥＲＶＩＣＥ＿ＤＡＴＡは，終端プロキシ間でアプリケーション・データを伝送するために使用される。その次の２つのメッセージは，アプリケーション・データの管理，すなわちフロー制御に使用される。次の３つのメッセージは，アプリケーションが必要としなくなったトンネル資源のクリーンアップを扱うために使用される。最後に、最終メッセージはエラー状態を管理するために使用される。

【００３０】この「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）は，トンネル資源管理とライフ・サイクルを支援するために開発された。当業者なら，同じ目的を達成するために同様の特徴と機能を含む他のプロトコルを作成することもできることがわかるであろう。

【００３１】終端プロキシの流れ図を図７に示す。このプロセスは機能ブロック７０１で中間プロキシに接続することによって開始される。次に，機能ブロック７０２で終端プロキシは自分のＣＯＭＭＯＮＵＮＩＱＵＥ＿ＩＮＦＯ及びＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥセットアップ情報を中間プロキシに送る。次に，機能ブロック７０３で終端プロキシは他のプロキシのＣＯＭＭＯＮＵＮＩＱＵＥ＿ＩＮＦＯ及びＴＯＰＯＬＯＧＹ＿ＥＸＣＨＡＮＧＥセットアップ情報を中間プロキシから受け取

る。機能ブロック704で、TOPOLOGY_EXCHANGEセットアップ情報に基づいてマスタ終端プロキシが選定される。判断ブロック705でこのプロキシがマスタ終端プロキシであると判断された場合、機能ブロック706でセキュリティ・ハンドシェークが開始される。そうでない場合には、機能ブロック707でこの終端プロキシはセキュリティ・ハンドシェークを待つ。機能ブロック708でセキュリティ・ハンドシェークが完了すると、機能ブロック709で終端プロキシはセキュリティ保護された接続を介してCOMMONUNIQUE_INFO及びTOPOLOGY_EXCHANGEセットアップ情報を再送信する。次に、機能ブロック710でプロキシはセキュリティ保護された接続を介して再びCOMMONUNIQUE_INFO及びTOPOLOGY_EXCHANGEセットアップ情報を受け取る。判断ブロック711での判断によりこの終端プロキシがクライアント終端プロキシである場合、機能ブロック712でこのプロキシはローカル・クライアント・アプリケーションが接続するのを待つ。機能ブロック713でローカル・クライアント・アプリケーションが接続されると、機能ブロック714で「軽セキュリティ保護トンネル・プロトコル（LSTP）を使用してその接続がセットアップされ、管理される。それに対して、判断ブロック711での判断によりこのプロキシがサーバ終端プロキシである場合、機能ブロック715でプロキシは他方の終端プロキシが接続を要求するのを待つ。機能ブロック716で他方の終端プロキシからCONNECTION_REQUESTメッセージを受け取ると、「軽セキュリティ保護トンネル・プロトコル（LSTP）を使用してその接続がセットアップされ、管理される。

【0032】中間プロキシの流れ図を図8に示す。このプロセスは、判断ブロック801で近隣プロキシからの新しい接続がないかどうかを検査することによって始まる。新しい接続がある場合、機能ブロック802でその新しい接続が受け入れられ、判断ブロック803で、前の接続からの一致するCOMMONUNIQUE_INFOメッセージが記憶されているかどうかを判断する。記憶されていない場合、機能ブロック804でCOMMONUNIQUE_INFO及びTOPOLOGY_EXCHANGEメッセージが中間プロキシに記憶され、プロセスは判断ブロック801に戻り、新しい接続を待つ。新しい接続を受信したとき、判断ブロック803での判断によって受信したCOMMONUNIQUE_INFOが前に記憶されていたCOMMONUNIQUE_INFOと一致する場合、機能ブロック805でその2つの接続を対にして2つの近隣プロキシ間に経路を確立する。一致したCOMMONUNIQUE_INFOは機能ブロック806で記憶域から除去される。機能ブロック807で、COMMONUNIQUE_INFO及びTOPOLOGY_EXCHANGEセットアップ情報が2つの新たに一致した近隣プロキシ間で交換され、判断ブロック801に戻る。ここで、近隣プロキシからの新しい接続ではないメッセージを受信した場合、判断ブロック808で、着信データを持つ既存の接続対が処理を待っていないかどうかを調べる検査を行う。そのような接続対がない場合は判断ブロック801に戻る。ある場合は、機能ブロック809で中間プロキシはバス・スルーとして機能する。すなわち、中間プロキシは一方の接続から着信データを受け取り、そのデータを接続された終端プロキシの対の他方の接続で送出する。

【0033】説明を簡単にするために1つの中間プロキシを示した。しかし、複数の中間プロキシをカスケードして、複数のファイアウォールを通る単一のトンネルを構築することもできる。この場合、それらの中間プロキシのうちのいくつかを、第2の着信接続を待つのではなく近隣中間プロキシとの接続を確立するように構成する。代替実施例では、1つのファイアウォールをアドレスする。これを図9に示す。この場合、A社の私設ネットワークまたはイントラネット21はXサーバ211と、ファイアウォール23の背後にあるサーバ終端プロキシ213とを含む。Xクライアント27はXサーバ211上で稼働しているアプリケーションをアドレスしようとしている。結合クライアント終端兼中間プロキシ28を使用してこの接続を容易にする。同様にして、図9はXクライアントとXサーバの役割が逆転している場合を考えることができる。

【0034】本発明のこの代替実施例による解決策では、2つのプログラムを使用する。一方のプログラムはX Windows Systemクライアントのプロキシとして機能し、他方のプログラムはX Windows Systemサーバのプロキシとして機能する。クライアント側プロキシは外部ネットワーク（すなわちインターネット）上で稼動し、したがってX Windows Systemクライアントによってアドレスされることができる。この外部クライアント側プロキシは、ファイアウォールを通って内部のサーバ側プロキシによってもアドレスされることができる。これは、ほとんどのファイアウォールがSOCKSパッケージを使用して、内部SOCKS認識プログラムが外部サービスをアドレスして接続するためのゲートウェイを提供しているためである。このプロセスは大体前述の通りである。

【0035】図10に示すように、さらに増強することによって、FTP、TELNET、HTTP及びSMTPなどのインターネット・プログラムの追加のSOCKS認識版が、セキュリティ保護されたトンネルを通って1つまたは複数のファイアウォールをナビゲートできるようになる。この改良によって、SOCKSサーバの機能がトンネル全体に付加され、それによって既存のSOCKS認識TCP/IPクライアントはトンネルの他

端にあるサーバと通信することができ、トンネル自体が2つ以上のファイアウォールをナビゲートすることがある場合であっても、トンネル自体を単一のファイアウォールとして扱うことができる。中間プロキシ26がバススルーとして機能し始めた後、サーバ終端プロキシ223とクライアント終端プロキシ213との間でLSTPが使用される。この実施例では、中間プロキシ26と2つの終端プロキシ213及び223は、SOCKSサーバ261を構成する。2つの終端プロキシは、SOCKS認識クライアント・アプリケーションからの送信を監視し、そのアプリケーションと通信するようにカスタマイズする必要がある。

【0036】RTELNET、RFTP、RHTTP（接頭字「R」は通常クライアントがSOCK化されていることを示す）などのようにSOCK化された多くの既存のクライアントがある。これらのクライアントは、典型的にはファイアウォール上にあるSOCKSサーバとの接続を期待しており、したがってSOCKSサーバにSOCKSプロトコルを送信することによって開始し、SOCKSサーバが応答することを期待する。

【0037】トンネルにSOCKSサーバ機能を付加することによって、トンネル全体がSOCKSサーバのジョブを行うようになる。したがってユーザは、RTELNETを使用してファイアウォールに接続する代わりに、RTELNETを使用してSOCKS使用可能トンネルに接続し、そのトンネル・インスタンスが通過する数のファイアウォールを通過することができる。この手続きのもう一つの考え方は、トンネルの機能に1つのプロトコル（すなわちSOCKS）を付加し、TELNET、ETP、HTTPなどのプロトコル群全体を獲得することである。SOCKSがなければ、TELNET、FTP、HTTPなどを認識して応答するようにトンネルを修正する必要が生じることになる。さらに、非SOCKS認識クライアントがSOCKSサーバに対して要求を行うことができるようにする市販の製品を使用して、どのようなクライアント・アプリケーションでもSOCKS認識終端プロキシにアクセスできるようにすることができる。

【0038】以上、本発明についてUNIXクライアント・ステーション上で稼働する場合について説明した。すなわち、サーバ、クライアント、終端プロキシ、及び中間プロキシはすべてUNIXワークステーション上で実施されている。しかし、たとえば、特定の用途や使用可能な資源によってはプロキシがパーソナル・コンピュータ（PC）またはメイン・フレーム・コンピュータ上で稼働できない理由はない。これは、UNIX基盤から別のプラットフォームにコードを移植するだけのことである。一般に、各プロキシが異なるハードウェア／ソフトウェア・プラットフォーム上で稼働しており、終端プロキシ、中間プロキシ、及び他の終端プロキシがトンネルを構成することができる。終端プロキシと中間プロキシは両方とも、それらが稼働しているコンピュータの資源を比較的要求しない。

【0039】まとめとして、本発明の構成に関して以下の事項を開示する。

【0040】（1）サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1のネットワークと第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む、第1のファイアウォールと、相互にアドレス可能なサーバ終端プロキシ及びサーバ・アプリケーションと、相互にアドレス可能なクライアント終端プロキシ及びクライアント・アプリケーションと、第1のファイアウォールの外部にあり、第1のネットワークと第2のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含み、サーバ終端プロキシとクライアント終端プロキシがそれぞれ第1のファイアウォールを介して中間プロキシとの接続を行い、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にバス・スルー通信トンネルを確立する、パケット交換ネットワーク通信システム。

（2）第1のネットワークと第2のネットワークのうちの他方のネットワークのコンピュータ資源を保護し、第2のファイアウォールが第2のファイアウォールの内部から外部への接続を行うことができるようにするソフトウェア・アプリケーションを含む第2のファイアウォールをさらに含む、上記（1）に記載のパケット交換ネットワーク通信システム。

（3）サーバ終端プロキシと、クライアント終端プロキシと、中間プロキシがSOCKSサーバ機能を有するトンネルを構成し、トンネル全体がSOCKSサーバのジョブを実行する、上記（2）に記載のパケット交換ネットワーク通信システム。

（4）サーバ・アプリケーションを稼働させる少なくとも1つのサーバを含む第1のネットワークと、クライアント・アプリケーションを稼働させる少なくとも1つのクライアントを含む第2のネットワークと、第1と第2のネットワークのうちの一方のネットワークのコンピュータ資源を保護し、第1のファイアウォールが第1のファイアウォールの内部から外部に接続を行うことができるようにするソフトウェア・アプリケーションを含む第1のファイアウォールと、サーバ・アプリケーションによるアドレスが可能なサーバ終端プロキシと、クライアント・アプリケーションによるアドレスが可能なクライアント終端プロキシと、第1のファイアウォールの外部

にあり、第１のネットワークと第２のネットワークの間の非信頼ネットワーク内にある中間プロキシとを含むパケット交換ネットワーク通信システムにおいて、第１のファイアウォールを介して中間プロキシにサーバ終端プロキシとクライアント終端プロキシとを接続し、中間プロキシがサーバ終端プロキシからの接続とクライアント終端プロキシからの接続とを接続してクライアントとサーバの間にパススルー通信トンネルを確立する方法であって、中間プロキシを始動させ、終端プロキシからの第１の接続を待つステップと、クライアント終端プロキシを始動させ、中間プロキシにクライアント・セットアップ情報を送信することによって中間プロキシとの接続を聞くステップと、中間プロキシによって、終端プロキシ・セットアップ情報を記憶し、その後で第２の接続を待つステップと、サーバ終端プロキシを始動させ、中間プロキシに終端プロキシ・セットアップ情報を送信することによって中間プロキシとの接続を聞くステップと、中間プロキシによって、クライアント終端プロキシの接続とサーバ終端プロキシの接続とを対にし、サーバ及び中間プロキシ・セットアップ情報をクライアント終端プロキシに送信し、クライアント及び中間プロキシ・セットアップ情報をサーバ終端プロキシに送信するステップと、その後で中間プロキシがクライアント終端プロキシとサーバ終端プロキシとの間のパス・スルーとして機能するステップとを含む方法。

（５）中間プロキシによってクライアント終端プロキシとサーバ終端プロキシの間の接続を対にした後で、クライアント終端プロキシとサーバ終端プロキシによってセキュリティ・ハンドシェークを交換するステップと、セキュリティ保護された回線で中間プロキシを介してクライアント終端プロキシとサーバ終端プロキシの間でセットアップ情報を再び交換するステップとをさらに含む、上記（４）に記載の方法。

（６）トンネルを介したクライアントとサーバの間のデータ交換が完了したときにクライアント終端プロキシとサーバ終端プロキシとの間のトンネル資源を解放するステップをさらに含む、上記（５）に記載の方法。

【図面の簡単な説明】

【図１】サーバがファイアウォールの背後にある場合の、クライアント・アプリケーションとサーバ・アプリケーションとの間の典型的な対話を示すブロック図である。

【図２】クライアントがファイアウォールの背後にある場合の、クライアント・アプリケーションとサーバ・アプリケーションとの間の典型的な対話を示すブロック図である。

【図３】それぞれがファイアウォールの背後にあるネットワークを有する２つの会社または組織の間の典型的なネットワーク構成を示すブロック図である。

【図４】２つの会社または組織の間にセキュリティ保護された通信チャネルまたはトンネルを構築するために本発明により使用する３つのタイプのプロキシを示すブロック図である。

【図５】図４に示すクライアント・プロキシ、中間プロキシ、及びサーバ・プロキシ間の対話を示すデータ流れ図である。

【図６】本発明の好ましい実施例による、「軽セキュリティ保護トンネリング・プロトコル」（ＬＳＴＰ）の概要を示す表である。

【図７】終端プロキシ上で実行されるプロセスを示す流れ図である。

【図８】中間プロキシ上で実行されるプロセスを示す流れ図である。

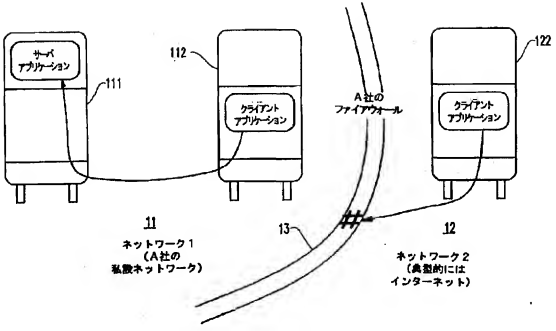
【図９】サーバ・プロキシが単一のファイアウォールを介してクライアント・プロキシに接続する本発明の代替実施例を示すブロック図である。

【図１０】ＳＯＣＫＳ認識クライアントが２つのファイアウォールを介したセキュリティ保護通信を可能にする本発明の他の代替実施例を示すブロック図である。

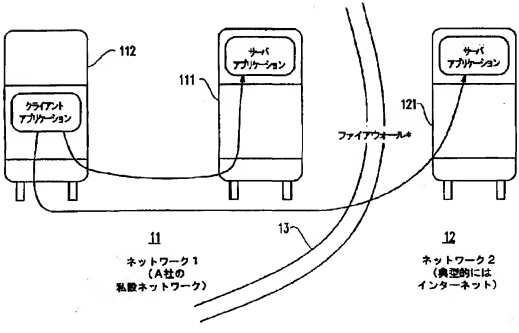
【符号の説明】

２１　ネットワーク
２２　ネットワーク
２３　ファイアウォール
２５　ファイアウォール
２６　中間プロキシ
２１１　Ｘサーバ
２１３　サーバ終端プロキシ
２２２　Ｘクライアント
２２３　クライアント終端プロキシ

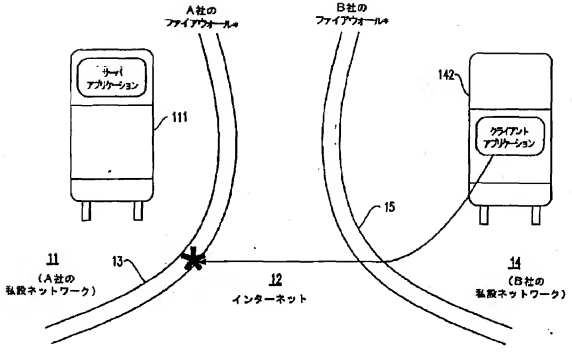
【図１】


サーバ
アプリケーション
111
112
クライアント
アプリケーション
A社の
ファイアウォール
122
クライアント
アプリケーション
11
ネットワーク１
（A社の
私設ネットワーク）
13
12
ネットワーク２
（典型的には
インターネット）

【図２】


112
クライアント
アプリケーション
111
サーバ
アプリケーション
ファイアウォール
121
サーバ
アプリケーション
11
ネットワーク１
（A社の
私設ネットワーク）
13
12
ネットワーク２
（典型的には
インターネット）

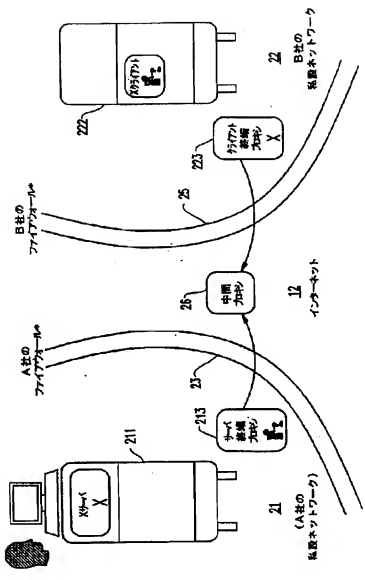
【図3】



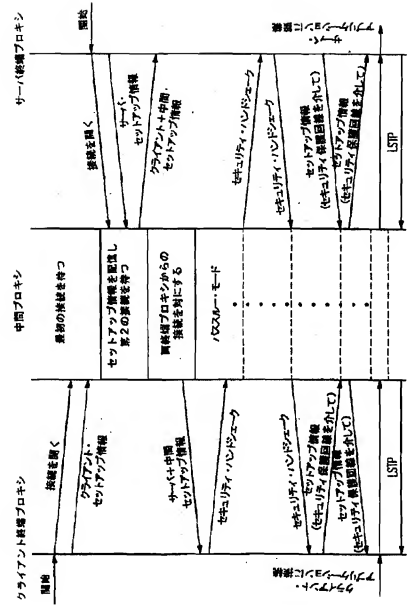
【図6】

LSTPメッセージ	メッセージの一般的目的
COMMONUNIQUE_INFO TOPOLOGY_EXCHANGE PROPERTY_EXCHANGE	トンネルセットアップと管理を扱う
CONNECTION_REQUEST CONNECTION_ACK CONNECTION_NACK SERVICE_BEGIN_REQUEST SERVICE_BEGIN_ACK SERVICE_BEGIN_NACK	トンネルを使用するアプリケーションの必要に応じて既存のトンネルの資源を管理する
SERVICE_DATA	終端プロキシ間でアプリケーション・データを伝達する
SERVICE_DATA_PAUSE SERVICE_DATA_RESUME	アプリケーション・データを管理するー フロー制御
SERVICE_FREE_REQUEST SERVICE_FREE_RESPONSE SHUTDOWN	アプリケーションが必要としなくなったトンネル資源のクリーンアップを管理する
ERROR	エラー条件を管理する

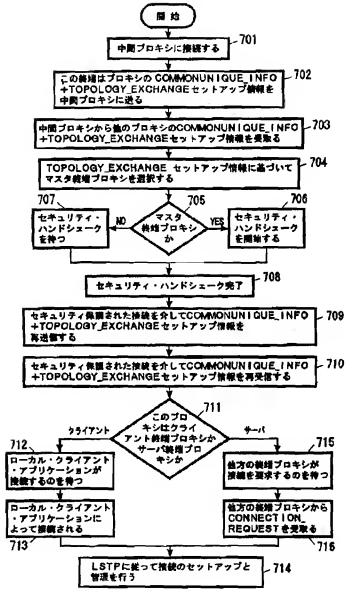
【図４】


B社の
私設ネットワーク
22
225
223
25
B社の
ファイアウォール
26
12
インターネット
23
A社の
ファイアウォール
213
211
21
(A社の
私設ネットワーク)

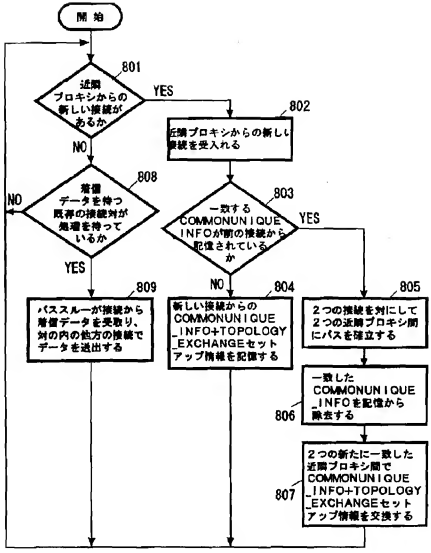
[図5]


クライアント終端プロキシ
中間プロキシ
サーバ終端プロキシ
開始
接続を開く
クライアント・セットアップ情報
最初の接続を待つ
セットアップ情報を記憶し第2の接続を待つ
接続を開く
サーバ・セットアップ情報
サーバ中間セットアップ情報
両終端プロキシからの接続を列にする
パススルー・モード
クライアント中間セットアップ情報
セキュリティ・ハンドシェーク
セキュリティ・ハンドシェーク
セットアップ情報
(セキュリティ情報は含んで)
セットアップ情報
(セキュリティ情報は含んで)
セットアップ情報
(セキュリティ情報は含んで)
セキュリティ・ハンドシェーク
LSTP
クライアントに接続
サーバに接続
クライアントへ
サーバへ

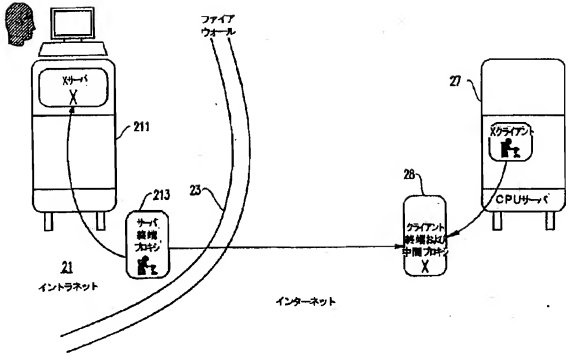
【図7】


開始
中間プロキシに接続する
701
この終端はプロキシのCOMMONUNIQUE_INFO
+TOPOLOGY_EXCHANGEセットアップ情報を
中間プロキシに送る
702
中間プロキシから他のプロキシのCOMMONUNIQUE_INFO
+TOPOLOGY_EXCHANGEセットアップ情報を受取る
703
TOPOLOGY_EXCHANGE セットアップ情報に基づいて
マスタ終端プロキシを選択する
704
705
マスタ
終端プロキシ
か
NO
YES
707
セキュリティ・
ハンドシェーク
を待つ
706
セキュリティ・
ハンドシェーク
を開始する
セキュリティ・ハンドシェーク完了
708
セキュリティ保護された接続を介してCOMMONUNIQUE_INFO
+TOPOLOGY_EXCHANGEセットアップ情報を
再送信する
709
セキュリティ保護された接続を介してCOMMONUNIQUE_INFO
+TOPOLOGY_EXCHANGEセットアップ情報を再受信する
710
711
このプロ
キシはクライ
アント終端プロキシか
サーバ終端プロ
キシか
クライアント
サーバ
712
ローカル・クライアント
・アプリケーションが
接続するのを待つ
715
他方の終端プロキシが
接続を要求するのを待つ
ローカル・クライアント
・アプリケーションに
よって接続される
713
他方の終端プロキシから
CONNECTION_
REQUESTを受取る
716
LSTPに従って接続のセットアップと
管理を行う
714

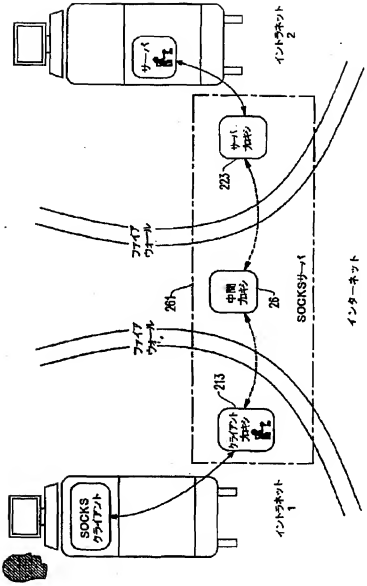
【図８】


開始
801
近隣プロキシからの新しい接続があるか
YES
NO
802
近隣プロキシからの新しい接続を受入れる
803
一致するCOMMONUNIQUE_INFOが前の接続から記憶されているか
YES
NO
804
新しい接続からのCOMMONUNIQUE_INFO+TOPOLOGY_EXCHANGEセットアップ情報を記憶する
805
２つの接続を対にして２つの近隣プロキシ間にパスを確立する
806
一致したCOMMONUNIQUE_INFOを記憶から除去する
807
２つの新たに一致した近隣プロキシ間でCOMMONUNIQUE_INFO+TOPOLOGY_EXCHANGEセットアップ情報を交換する
808
着信データを待つ既存の接続対が処理を待っているか
NO
YES
809
パススルーが接続から着信データを受取り、対の内の他方の接続でデータを送出する

【図９】


ファイア
ウォール
Xサーバ
211
213
23
サーバ
接続
プログラム
21
イントラネット
27
Xクライアント
28
CPUサーバ
クライアント
接続
中継プログラム
インターネット

【図１０】



フロントページの続き

(72)発明者　ピーター・エフ・ガーヴィン
アメリカ合衆国12487　ニューヨーク州アルスター・パーク　ハルスタイン・レーン　8

(72)発明者　ジェフリー・ダブリュ・スタテン
アメリカ合衆国12603　ニューヨーク州ポーキープシー　ハッケンサック・ロード　257

(72)発明者　ワイキ・エル・ライト
アメリカ合衆国　94086　カリフォルニア州サニーベイル　イースト・イブリン・アベニュー825　アパートメント・ナンバー212